UUV On-Board Path Planning in a Dynamic Environment for the Manta Test Vehicle

Piero Miotto
Draper Laboratory
Aerospace Control Group
555 Technology Square
Cambridge, MA, 02139, US
pmiotto@draper.com

John Wilde
Draper Laboratory
Decision Systems
555 Technology Square
Cambridge, MA, 02139, US
jwilde@draper.com

Alberico Menozzi
Naval Undersea Warfare Center
Division Newport Code 8231, B1302/2
1176 Howell Street,
Newport, RI, 02841
menozzia@npt.nuwc.navy.mil

***Abstract* - This paper presents recent work in the areas of simulation, mission planning, and mission execution for an unmanned undersea vehicle (UUV). The UUV we consider is the Manta Test Vehicle (MTV), operated by the Naval Undersea Warfare Center (NUWC) in Newport, RI. A 6-Dof Simulink model of the MTV vehicle dynamics augmented with an autopilot is used to test the algorithms.**

The on-board mission planner generates reference trajectories for the vehicle to follow, taking into consideration bathymetry data and moving obstacles that are within the forward-looking sonar range. A trajectory consists of a sequence of waypoints and associated headings from the current vehicle location and orientation to the goal. Trajectory generation takes into consideration the dynamic capabilities of the MTV. The D* algorithm - an extension to the Dijkstra shortest-path algorithm which allows efficient re-planning when arc-costs change - is used to generate and maintain a safe trajectory. Trajectory re-planning is triggered when the sonar detects an obstacle in the trajectory currently being followed.

A Model Predictive Control (MPC) algorithm is inserted between the D* algorithm and the vehicle inner loop autopilot. The MPC algorithm issues the reference commands to the autopilot to allow the vehicle to follow the planned trajectory. The cost function within the MPC algorithm can be changed depending on the guidance task. The MPC algorithm uses a full nonlinear model of the MTV vehicle to project ahead the output trajectory and employs orthogonal Laguerre polynomials to create basis functions that are used in the synthesis of reference commands to the autopilot. The MPC controller also provides a second layer of obstacle avoidance capability and keeps the vehicle on-track in the presence of a current.

I. INTRODUCTION

Since 1996 the US Navy's Naval Undersea Warfare Center (NUWC) has been developing a concept for undersea warfare in the new century. Playing a major role in this futuristic concept are mission-reconfigurable unmanned undersea vehicles (MRUUVs) that will be capable of carrying out surveillance, tactical oceanography, mine warfare, and anti-submarine warfare missions. The Manta Test Vehicle (MTV) is the first prototype.

The missions require UUVs to operate with contacts present for extended periods of time in complex, partially known environments. The ability to dynamically plan vehicle paths to avoid dynamic contacts and create safe vehicle trajectories in complex shallow water (SW) environments is extremely important for mission success and overall system survivability. Due to the types of missions, the expected complexity of the environment, and changing tactical situations, significant levels of re-planning are expected to be performed during the mission. Additionally, Target Motion Analysis (TMA) maneuvers must be conducted while balancing route following, obstacle avoidance, and contact avoidance concerns.

Many algorithms for path planning/generation used in robotics research programs have been designed to function in static environments. Due to the dynamic nature of the anticipated operational environments, alternate algorithms are being investigated by Draper Laboratory and NUWC.

This paper describes how a D* algorithm, a B-Spline curve generator, and a Model Predictive Control algorithm can be integrated to form an on-board path planning, guidance, and control system.

II. UNDERWATER VEHICLE MODEL

The motion of underwater vehicles is analyzed by considering two coordinate frames as shown in Fig. 1. The kinematics of the body-fixed frame, **O**, is expressed relative to the earth-fixed frame, $\mathbf{O}_E$, which is considered inertial. The MTV dynamics model presented here follows the notation of [1] and [2], and assumes forward motion with unidirectional propeller rotation.

The mathematical framework used is the standard set of six-degrees-of-freedom (6-Dof) equations of motion for a rigid body. The state vectors $\boldsymbol{\nu} = [u\ v\ w\ p\ q\ r]^T$ and $\boldsymbol{\eta} = [x\ y\ z\ \phi\ \theta\ \psi]^T$ are

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
01 SEP 2003	**N/A**	**-**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
UUV On-Board Path Planning in a Dynamic Environment for the Manta Test Vehicle	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Draper Laboratory Aerospace Control Group 555 Technology Square Cambridge, MA, 02139, US**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002146. Oceans 2003 MTS/IEEE Conference, held in San Diego, California on September 22-26, 2003. U.S. Government or Federal Purpose Rights License., The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**UU**	**8**	

used to describe the vehicle's dynamics: $\boldsymbol{\nu}$ is a vector of linear and angular rates in the body-fixed frame; (x, y, z) is the position of the vehicle's origin relative to the earth-fixed frame; ϕ, θ, and ψ are the roll, pitch, and yaw angles (i.e., the Euler angles).

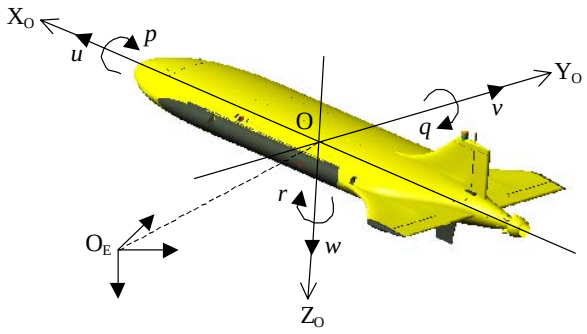

Fig. 1 NUWC's Manta Test Vehicle (MTV) and coordinate frame definitions.

The control input vector is $\mathbf{u} = [n \ \delta_a \ \delta_e \ \delta_r]^T$, where n>0 is the speed of the propeller, and δ_a, δ_e, δ_r are the angular positions of the aileron, elevator, and rudder control surfaces. With these definitions, the state equations are

$$\dot{\boldsymbol{\nu}} = \mathbf{M}^{-1}\left[\mathbf{h}(\boldsymbol{\nu}) - \mathbf{C}_{RB}(\boldsymbol{\nu})\boldsymbol{\nu} - \mathbf{g}(\boldsymbol{\eta}) + \boldsymbol{\tau}(\boldsymbol{\nu},\mathbf{u}) + \boldsymbol{\tau}_E\right], \quad (1)$$

$$\dot{\boldsymbol{\eta}} = \mathbf{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \quad (2)$$

where $\mathbf{M}$ is the inertia matrix (including the inertia of the water surrounding the vehicle), $\mathbf{h}(\boldsymbol{\nu})$ is a vector of hydrodynamic forces and moments due to body rates only, $\mathbf{C}_{RB}(\boldsymbol{\nu})$ is the rigid-body Coriolis and centripetal matrix, $\mathbf{g}(\boldsymbol{\eta})$ is the restoring forces and moments vector (i.e., the effects of gravity and buoyancy), $\boldsymbol{\tau}(\boldsymbol{\nu},\mathbf{u})$ is the vector of hydrodynamic forces and moments due to control surface deflections and propeller speed, and $\boldsymbol{\tau}_E$ represents environmental forces and moments (i.e., due to ocean currents, waves, etc.). $\mathbf{J}(\boldsymbol{\eta})$ is the body-to-earth kinematic transformation.

Each element of the vectors $\mathbf{h}(\boldsymbol{\nu})$ and $\boldsymbol{\tau}(\boldsymbol{\nu},\mathbf{u})$ is expressed as a linear combination of functions that are appropriately chosen to best describe the hydrodynamics of the vehicle. The coefficients of each linear combination and the hydrodynamic derivatives in $\mathbf{M}$ (i.e., the added-mass coefficients) are jointly called the hydrodynamic coefficients, and are determined experimentally and via numerical methods. Typically, several sets of hydrodynamic coefficients are produced, each corresponding to a certain operating condition of the vehicle. In the case of MTV, three sets of coefficients have been obtained: 2.5-knots, 5-knots, and 10-knots coefficients.

The mapping between the MTV's actual vehicle control surfaces, $\boldsymbol{\delta} = [\delta_{top} \ \delta_{stbd} \ \delta_{btm} \ \delta_{port}]^T$, and the "virtual" ones, $\boldsymbol{\delta}_v = [\delta_a \ \delta_e \ \delta_r]^T$, used in the 6-Dof equations of motion, is

$$\boldsymbol{\delta} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & -1 & 0 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix} \boldsymbol{\delta}_v . \quad (3)$$

Each actual control surface is driven through various joints and linkages by a high-performance permanent-magnet DC motor. Control surface displacement is achieved by controlling the armature voltage, u_a, applied to the motor windings.

The MTV's propeller is driven by a high-performance permanent-magnet synchronous motor. More details on the modeling and control of this motor can be found in [3]. The nonlinear torque load due to the hydrodynamics of the propeller is

$$Q(n,V_a) = Q_{nn}n^2 + Q_{nVa}nV_a , \quad (4)$$

where Q_{nn}>0 and Q_{nVa}<0 are design parameters that depend on propeller geometry and other hydrodynamic variables; V_a is the speed of the water going into the propeller, which is typically a fraction of the vehicle's forward speed, u. Similar to the control surface dynamics, electromechanical constraints impose limits on the ranges of u_a and n, and these constraints are included in the vehicle model.

The MTV sensor suite is composed of many subsystems, including a strapdown ring-laser gyro Inertial Navigation System (INS), a phased-array Doppler Velocity Sonar (DVS), a depth sensor, and a differential GPS receiver. Sensor fusion based on Kalman filtering is performed to deliver measurements of every state variable of interest (i.e., $\boldsymbol{\nu}$, $\boldsymbol{\eta}$, $\boldsymbol{\delta}$, and n). All the available information on sensor dynamics, filtering, quantization values, dynamic ranges, noise power levels, etc., has been incorporated in the nonlinear model.

III. PATH PLANNING AND CONTROL

A conventional vehicle control system has two independently designed subsystems, a guidance system and a flight control system. Traditionally the guidance is responsible for generating suitable guidance commands for the inner loop autopilot. The autopilot, also called the flight control system,

keeps the vehicle on the trajectory commanded by the guidance algorithm.

In the current design there is not such a clear distinction between the two functions. In general the D* and the B-Spline algorithms can be viewed as the guidance function and the MPC and the inner loop autopilot as the control function. Fig. 2 shows the Simulink implementation of the MTV control system described in this paper.

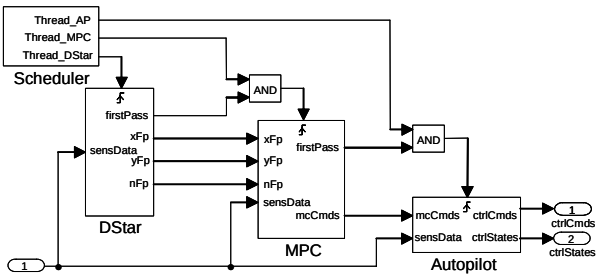

Fig. 2: Vehicle Control System

The path planning and control scheme presented in this paper is divided in 3 stages.

In the first stage a minimum path is computed by a D* algorithm from the current UUV position to the target position based on bathymetry data and the current location of the obstacles. The path generated by the D* algorithm cannot be directly tracked by the control function because of the sharp corners connecting consecutive segments.

The second stage uses a B-Spline algorithm to generate a trajectory that the UUV can fly over the next *N* waypoints. The optimum B-Spline path is generated incrementally as the vehicle proceeds along the trajectory.

The third stage implements a Model Predictive Control algorithm to generate the steering commands that keep the UUV on the desired trajectory.

The three stages are described in detail in the next sections.

A. D Planning Algorithm*

The responsibility of the mission planner is to maintain a list of future waypoints for the MPC algorithm to follow. These waypoints must define a path which the test vehicle is capable of tracking and which allows the vehicle to travel at its commanded depth while avoiding collisions with moving obstacles.

Our path planning algorithm operates on a 5 km square map of bathymetry data (divided into 50m cells) for a section of the Narragansett Bay, RI. For the purposes of path planning, the vehicle state is defined by its cell and one of eight possible headings. The legal state transitions are shown in Fig. 3. From any given state there is a maximum of three neighboring states.

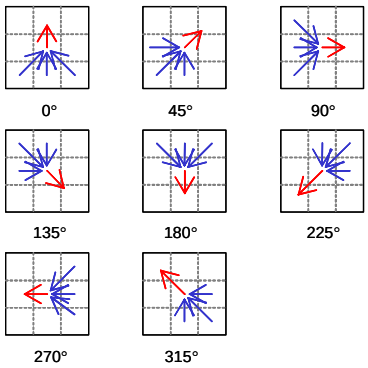

Fig. 3 State Transitions

Our choice of map cell-size and legal state transitions enables a minimum vehicle-turn radius constraint to be met. Past applications used a finer heading resolution, allowing state transitions in increments of 10 degrees. This leads to a much larger search space. Using the MPC approach together with a spline generation algorithm allows a smaller search space while still generating feasible trajectories.

The cost of making a state transition, also called the arc-cost, is the distance from the center of the current cell to the center of the next cell. This performance index defines a shortest-path type problem, but could easily be modified to include other costs.

A C++ implementation of the D* algorithm [4] is used to find the optimum path through the map. The algorithm is an extension of Dijkstra's shortest-path algorithm giving the advantage of efficient re-planning if arc-costs change during traversal of the path. Rather than re-solving the entire shortest-path problem when some arc-costs change, a reduced set of states are re-opened and the algorithm determines the (possibly new) minimum-cost path from its current position to the goal.

Obstacle Avoidance

The vehicle's sensors can determine the traversability of any cell within a specified distance from the current vehicle position. If an obstacle is detected along the vehicle's planned path, the map is updated and the re-planning function is called. In Fig. 4 an obstacle is shown crossing the vehicle's planned path. However, given the limited vehicle sensor range this obstacle goes undetected and no replanning is performed.

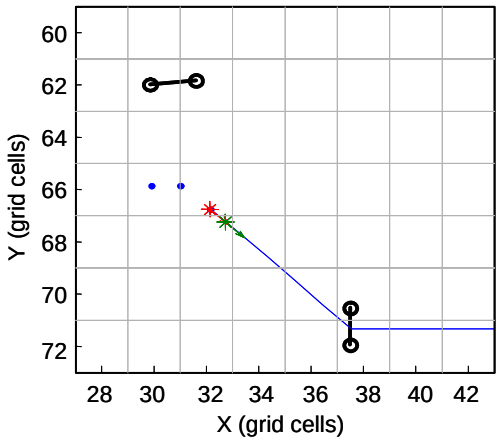

Fig. 4 Undetected Obstacle

In Fig. 5, the vehicle has moved forward, bringing the obstacle within sensor range. The obstacle is detected and the map is updated to reflect the cells that are now invalid (X=37 and Y= 70,71,72). The planner strategy is to conduct a simple re-plan the first time an obstacle is detected. The vehicle then follows the new path. In Fig. 5 we see the vehicle following the new path passing in front of the obstacle, the dotted line represents the old path.

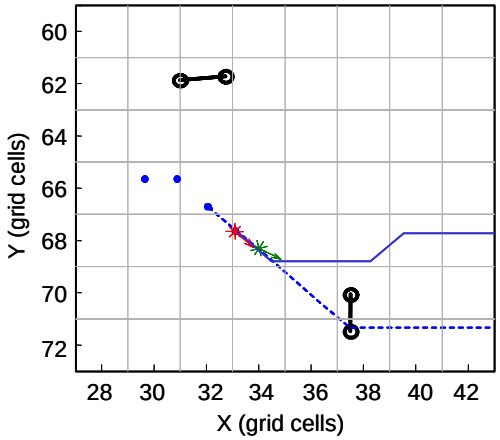

Fig. 5 Obstacle Detected. Simple Replan

However, if the new path becomes blocked again by the same obstacle (in case of a dynamic obstacle), an alternate path is planned. Different strategies can be implemented to evade a blocking dynamic obstacle. Fig. 6. shows a simple strategy that brings the vehicle behind the obstacle. Once the point behind the vehicle is reached a new path to the target is calculated.

One of the major concerns in using the D* algorithm in a real time application is computation speed. Tab. 1 shows the processing time required for these scenarios, running on a pentium IV (2.2GHz). The D* algorithm is currently executed at 0.1 Hz, this gives a margin to increase the planning search space or the rate at which the algorithm is executed.

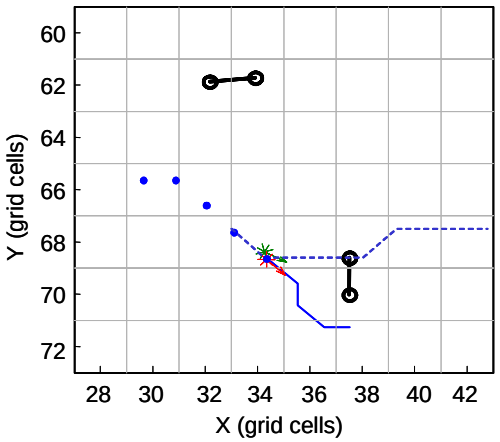

Fig. 6 Alternate plan behind obstacle

	Vehicle State's Considered	Time (s)
Initial Plan (Fig. 4)	19678	0.25
Replan (Fig. 5)	380	0.016
Alternate Plan (Fig. 6)	101	0.031
Final Plan	9280	0.109

Tab. 1 D* algorithm runtime

B. B-Spline Reference Trajectory

The trajectory calculated by the D* algorithm is a set of waypoints connected by path segments. Two adjacent segments can lie on the same line or at angle of ±45 degrees. Fig. 7 shows an hypothetical trajectory with 12 waypoints.

The B-Spline is a technique used in computer graphics to generate smooth curves. An introduction to spline curves can be found in reference [9].

The B-Spline algorithm is applied to 5 waypoints in the D* trajectory. The first control point is the waypoint just behind the current location of the vehicle. In Fig. 7 large filled circles indicate the spline control points, the X indicate past waypoints, and empty circles future waypoints. Fig. 7 also shows how the degree *T* of the spline affects the shape of the curve. For T = 1 the spline curves are just connecting the waypoints with straight segments, increasing the degree we get smoother curves between the waypoints. The distance between the waypoints and the degree of the spline are selected based on the dynamics of the vehicle.

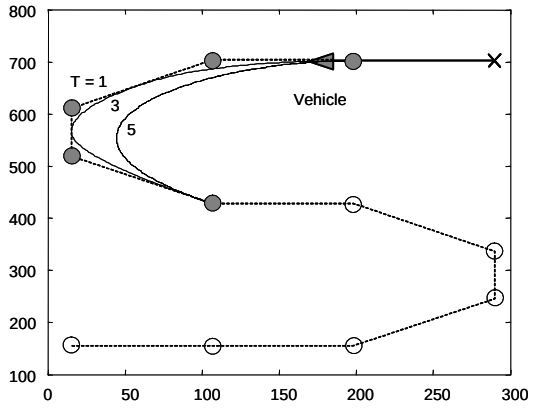

Fig. 7: B-Spline Curves

Fig. 8 shows the response of the MTV vehicle to a step command in the y (North) direction. The vehicle has difficulties tracking low degree B-Spline (order 1 or 2), while it is able to well track B-Splines of order 3 and 4. Higher order splines generate smoother trajectories and can be used in areas where it is not required to remain close to the segments connecting the waypoints. The order of the spline can be changed by the autopilot during the mission.

Once the B-Spline curve over the next 5 waypoints is generated it is passed to the MPC. The MPC is then responsible to track the trajectory as close as possible. In the MTV implementation the B-Spline algorithm is running at 2Hz, the same rate as the MPC and it is incorporated in the block called MPC in Fig. 2.

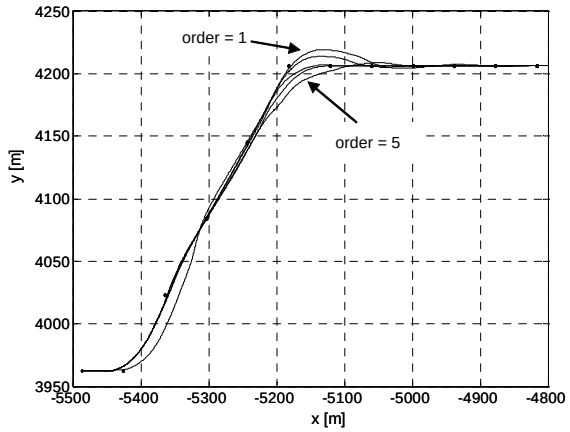

Fig. 8: Yaw Step Response for Different Spline Order

C. MPC Based Autopilot

The MPC is responsible for tracking the B-Spline reference trajectory by generating yaw rate commands for the inner loop autopilot. The reference trajectory is defined in terms of x (East) and y (North) positions on the map.

The MPC methodology is based on the on line solution of a finite horizon open loop control problem subject to plant dynamics and constraints on states, outputs, and inputs. In our application the plant is formed by the close loop autopilot-vehicle system.

Fig. 9 shows the basic principle of model predictive control.

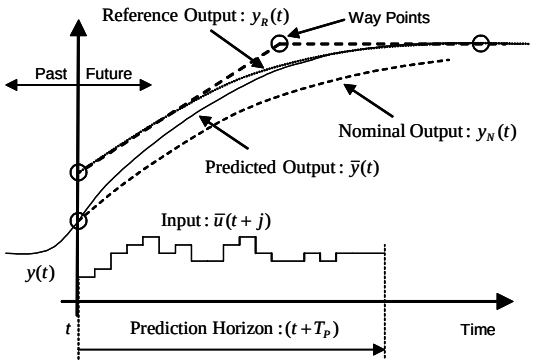

Fig. 9: Principle of Model Predictive Control

MPC is based on the following points:

- At each instant t, the future outputs $\bar{y}(t+j)$ are predicted over a determined finite horizon T_P. An internal model of the plant is used by MPC to predict the future output of the plant. In this application the internal model is nonlinear.

$$\begin{aligned} \dot{\bar{x}}(t) &= f(\bar{x}(t), \bar{u}(t)) \\ \bar{y}(t) &= g(\bar{x}(t), \bar{u}(t)) \end{aligned} \tag{5}$$

- The future control signals are calculated by optimizing a performance index to keep the process as close as possible to a reference trajectory $y_R(t)$. In the current application the index is a weighted quadratic function of the squared distance between the predicted output trajectory and the reference trajectory ($\bar{d}(\tau)$). The optimization problem is

$$\begin{aligned} &\min_{\bar{u}(\bullet)} J\left(y(t), \bar{u}(t); T_P\right) \\ &J\left(y(t), \bar{u}(t); T_P\right) = \int_t^{t+T_P} G(\bar{y}(\tau), \bar{u}(\tau) d\tau \\ &G(\bar{y}(\tau), \bar{u}(\tau)) = \bar{d}(\tau)^T Q \bar{d}(\tau) \end{aligned} \tag{6}$$

subject to equation (5) and the following constraints:

$$\begin{aligned} &\bar{u}(\tau) \in U, && \forall \tau \in [t,\ t+T_P] \\ &\bar{y}(\tau) \in Y, && \forall \tau \in [t,\ t+T_P] \end{aligned} \tag{7}$$

where T_P is the prediction horizon. *Q* in equation 6 is the weighting matrix. Q is a diagonal matrix that allows different weightings of present and future errors.

In equation 6 the squared distance between the reference trajectory and the predicted trajectory is defined as

$$\bar{d}(\tau) = \left(\bar{\xi}(\tau) - \xi_R(\tau)\right)^2 + \left(\bar{\eta}(\tau) - \eta_R(\tau)\right)^2 \quad (8)$$

where ξ and η are the North and East position respectively.

- At each sampling instants only the current optimum control signal is sent to the plant while all the future ones are neglected

$$u^*(t) = \bar{u}^*(\tau, y(t), T_P), \quad \tau \in [t,\ t+\delta] \quad (9)$$

where δ is the sampling time. This process is repeated at the next cycle.

In order to reduce the optimization problem defined in equation 6 to a linear quadratic optimization problem, the output predicted trajectory is expressed as a linear combination of variations from a nominal trajectory. The nominal trajectory is obtained holding the previous input $u(t-\delta)$ over the whole horizon T_P. This is also called the free response: the response of the pant when the previous input is held constant over the prediction horizon. The nominal input

$$\bar{u}_N(\tau) = u(t-\delta), \quad \forall \tau \in [t,\ t+T_P] \quad (10)$$

when applied to the plant model produces the nominal output trajectory

$$\bar{y}_N(\tau) \quad \forall \tau \in [t,\ t+T_P] \quad (11)$$

A set of polynomial basis functions, $b_i(\tau)$, are then added to the nominal input:

$$\bar{u}_i(\tau) = \bar{u}_N(\tau) + b_i(\tau), \quad \forall \tau \in [t,\ t+T_P] \quad (12)$$

Different basis functions can be used to perturb the nominal input. The stability and the convergence of the MPC control problem is strictly related to the type and the perturbation level of basis function selected. The most common basis functions are step input equally or logarithmically spaced over the prediction horizon. Although these basis gave good results we found that Laguerre orthonormal basis functions helped to form a numerically well posed LQ problem. The first few Laguerre polynomials are

$$L_0(\tau) = 1$$
$$L_1(\tau) = -\tau + 1$$
$$L_2(\tau) = (\tau^2 - 4\tau + 2)/2$$
$$L_3(\tau) = (-\tau^3 + 9\tau^2 - 18\tau + 6)/6$$

The results presented in this paper use a 12th order Laguerre polynomial basis. The response of the plant to the *i*th basis function over the prediction horizon is

$$\bar{y}_i(\tau) \qquad \forall \tau \in [t,\ t+T_P] \quad (13)$$

The difference between the *i*th basis output and the nominal output is

$$\bar{\varepsilon}_i = \bar{y}_i(\tau) - \bar{y}_N(\tau) \qquad \forall \tau \in [t,\ t+T_P]$$

Collecting all the basis input and output deviation from the nominal in matrix form, we have:

$$B = \begin{bmatrix} b_1 & b_2 & \dots & b_N \end{bmatrix}$$
$$E = \begin{bmatrix} \varepsilon_1 & \varepsilon_{21} & \dots & \varepsilon_N \end{bmatrix}$$

The output of the plant can be expressed as the sum of the nominal output and a linear combination of the basis output deviation from the nominal:

$$\bar{y}(\tau) = \bar{y}_N(\tau) + E\alpha \qquad \forall \tau \in [t,\ t+T_P]$$
$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix} \quad (14)$$

The cost function of the MPC control problem can then be expressed in the following form

$$\begin{aligned} J &= \left(\bar{y}(\tau) - y_R(\tau)\right)^T Q\left(\bar{y}(\tau) - y_R(\tau)\right) \\ &= \left(\bar{y}_N(\tau) + E\alpha - y_R(\tau)\right)^T Q\left(\bar{y}_N(\tau) + E\alpha - y_R(\tau)\right) \\ &= \left(E\alpha - (y_R(\tau) - \bar{y}_N(\tau))\right)^T Q\left(E\alpha - (y_R(\tau) - \bar{y}_N(\tau))\right) \\ &= \left(E\alpha - \varepsilon_R(\tau)\right)^T Q\left(E\alpha - \varepsilon_R(\tau)\right) \end{aligned} \quad (15)$$

Where $\varepsilon_R(\tau)$ is the difference between the reference trajectory and the free response. In our application the output is already the distance square from the reference trajectory and so $y_R(\tau)$ is equal to zero. The MPC is so reduced to a constrained LQ problem in alpha.

$$\min_{\alpha} J(\alpha) = \min_{\alpha} \frac{1}{2}\alpha^T H\alpha - F^T\alpha$$
$$\alpha \in A$$

where H and F can be easily calculated from equation 15.

At this point we assume a linear relationship between the basis input (*B*) and output (*E*). The

optimum α^* is used to calculate the optimum input at time *t*:

$$\bar{u}^*(t) = \bar{u}_N(t) + B(t)\alpha^* \quad (16)$$

As we mentioned earlier the optimum input is applied to the plant and the process is repeated at the next sampling time.

IV. APPLICATION RESULTS

The algorithms described in the previous section were implemented in C and C++, and integrated in a Simulink simulation of the MTV test vehicle. Fig. 2 shows the top level diagram of the Flight Control System Simulink subsystem. The Scheduler calls the various software tasks at their appropriate rates. Considering the slow dynamic response capability of the vehicle, the D* algorithm is called at 0.1 Hz, the B-spline/MPC controller at 0.5 Hz, and the inner loop autopilot at 25 Hz.

One of the key points in the design of the MPC controller is the definition of the prediction horizon. The horizon needs to be long enough to capture the slow dynamics that are relevant in the calculation of the cost function. The cost function is calculated based on the distance from a reference trajectory defined as North and East positions. The vehicle 90° turning time is a function of speed and it is of the order of 1 minute. Therefore, in order to track 90° turns a prediction horizon of at least 1 minute is necessary to formulate a well posed MPC problem.

The MPC controller uses a 2 Hz discrete model of the plant to predict the output trajectory. With one input control variable, the heading rate, and a prediction horizon of one minute the MPC optimum control problem has a search space of 120 variables (if no blocking is applied to the input). Fortunately the basis function approach reduces the size of the problem from 120 down to 12 independent bases, drastically reducing the complexity and the computation time.

The slow turning capability of the vehicle justifies the decision to run the D* algorithm at 0.1Hz. There is no benefit in updating the plan faster than 0.1 Hz because of the limited capability of the vehicle to avoid any obstacles that appear in close range.

In the simulation the MTV vehicle operates between 2 and 7 knots at a constant depth of 20 ft. The simulations were conducted using a map and bathymetry data of the Narragansett Bay.

An example of replanning in order to avoid a moving obstacle is shown in Fig. 10. The obstacle is moving North along the 2500 m East meridian. Several plans are calculated along way to generate alternate paths around the moving obstacle. Initially the vehicle tries a path in front of the obstacle, then due to its lower relative velocity, the D* algorithm switches to a path behind the obstacle.

Fig. 11 shows the trajectory followed by the MTV vehicle during a mission with 4 legs. The first leg brings the vehicle from the *Start* position to point A with a required heading of 180°, the vehicle next proceeds to points B (required heading 270°), C (225°), and D (180°) before returning to the *End* point. The circle around point C is necessary to achieve the 225° heading. The vectors in Fig. 11 represent the direction of a 3 knot current. A perfect measurement of the current is used in the MPC to calculate the optimum steering commands. Finally Fig. 12 shows the reference trajectory generated by the B-Spline algorithm to follow the waypoints out of a narrow inlet.

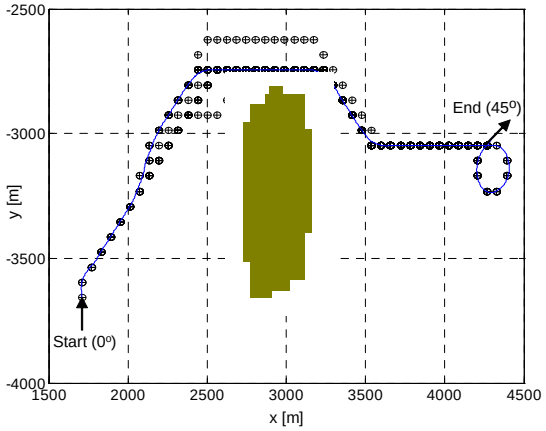

Fig. 10 Replanning for obstacle avoidance

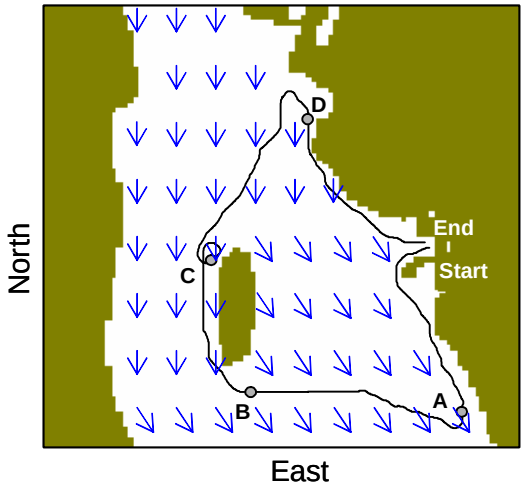

Fig. 11 Path with Multiple Objectives

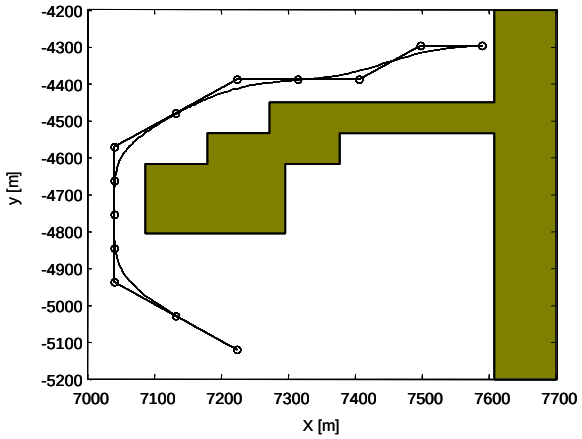

Fig. 12 Spline Fit for first leg of initial path

VI. CONCLUSIONS

This paper presents the results of a feasibility study for on-board path planning and vehicle control algorithms for the Manta Test Vehicle. The path planner, operating at 0.1 Hz, calculates waypoints that steer clear of obstacles in the environment. An inner-loop autopilot, running at a faster rate, generates vehicle commands based on MPC and spline-generation algorithms. Together, these approaches demonstrate the capability for real-time avoidance of moving obstacles, subject to the dynamic constraints of the test vehicle.

Future work involves exercising these algorithms under Draper Laboratory's high fidelity C simulation framework. The path planning algorithm will also be augmented with a method for determining a minimum distance path that meets a given probability measure for successful traversal.

Acknowledgements

The authors wish to acknowledge current sponsorship of this work by the Office of Naval Research, under the UUV Project, Autonomous Operations Future Naval Capability (AOFNC) program.

References

[1] T. I. Fossen, *Guidance and Control of Ocean Vehicles*, Wiley, 1994.

[2] SNAME, *Nomenclature for Treating the Motion of a Submerged Body Through a Fluid*, Tech. and Res. Bull. No. 1-5, April 1950.

[3] A. Menozzi, T. C. Gagliardi, and S. E. Lyshevski, "Dynamics and Control of MTV: a Multipurpose Unmanned Underwater Vehicle", *Proceedings of the American Control Conference*, Chicago, IL, 2000, pp. 70-74.

[4] Stentz A., "Optimal and Efficient Path Planning for Partially-Known Environments", *Proceedings IEEE International Conference on Robotics and Automation*, May 1994.

[5] Stentz A., "The focussed D* algorithm for real-time replanning", *Proceedings of the International Joint Conference on Artificial Intelligence*, August 1995.

[6] Stentz A., "Map-Based Strategies for Robot Navigation in Unknown Environments", *Proceedings AAAI 96 Spring Symposium on Planning with Incomplete Information for Robot Problems*, 1996.

[7] Maciejowski J. M., Predictive Control with Constraints, Prentice Hall, Pearson Education 2002.

[8] Judd K. B., and McLain T. W. "Spline based Path Planning for Unmanned Air Vehicles", AIAA-2001-4238.

[9] Chankin, G., "An algorithm for high speed curve generation", *Computer Graphics and Image Processing*, 3 (1974), pp. 346-349.